Implémentation des Plugins Logstash de Généralisation et de Chiffrement pour l'Anonymisation et la Pseudonymisation

Ali Hassan, Amine Mrabet, Patrice Darmon

Research & Innovation - Umanis
7, Rue Paul Vaillant Couturier, 92300 Levallois-Perret, France
{ahassan, amrabet, pdarmon} @umanis.com

Résumé. Ce papier présente une nouvelle implémentation des méthodes de protection des données à caractère personnel basées sur des algorithmes de chiffrement et de généralisation spécifiques mis en oeuvre dans des plugins Logstash. Notre algorithme de chiffrement est adapté aux données personnelles en considérant les différentes catégories : identifiants, quasi-identifiants et attributs sensibles. En outre, notre solution d'anonymisation propose plusieurs méthodes de généralisation, paramétrables selon les types des données. Afin de valider les résultats de ces méthodes, nous proposons également une étape de vérification des modèles de protection de la vie privée comme le k-anonymat et le l-diversité.

1 Introduction

Les besoins de collecte, de stockage et d'analyse des données sont en croissance constante notamment en raison de la révolution liée aux objets connectés. L'analyse de ces données est essentielle pour les entreprises avec différents enjeux : IA, statistique, publicité, etc... Cependant, le stockage et l'analyse de données personnelles posent des vrais problèmes de confidentialité. Les techniques de protection de la vie privée sont conçues et mises en oeuvre pour équilibrer les usages et la confidentialité des données à caractère personnel (DCP). La confidentialité et les principes de la protection des DCP doivent être garantis dans toutes les phases de collecte, stockage, traitement, analyse et partage des données.

Dans le cadre d'une démarche de protection des données, l'étape de découverte des données est indispensable. Une automatisation de cette étape est proposée dans (Mrabet et al., 2019). Les recommandations relatives à la pseudonymisation et à l'anonymisation visent à protéger l'individu et à renforcer la conformité au RGPD. Dans ce contexte, les techniques d'anonymisation sont une solution pour la protection, mais elles semblent inappropriées dans certaines circonstances. En outre, la pseudonymisation est utilisée à la fois pour réduire les risques de profilage et aider à respecter les obligations de protection des DCP. Donc, l'anonymisation et la pseudonymisation peuvent être utilisées de manière complémentaire ou séparée.

Motivation et cas d'étude

Dans le cadre d'un projet Big Data de smart territoire, une collectivité territoriale, qui collecte des données d'utilisation de wifi dans l'espace public, voudrait protéger le stockage (chez

un hébergeur) et analyser les données collectées (un extrait simplifié des données collectées se trouve dans la Table 1). L'analyse des données peut être associée à différents types de traitements : statistiques, clustering, etc... Un des besoins de ce projet est de remplacer chaque valeur de l'identifiant par une valeur pseudonymisée unique mais réversible. C'est pourquoi nous avons choisi le chiffrement pour réaliser la pseudonymisation. En outre, des informations concernant l'utilisation du wifi peuvent être mises à disposition en données ouvertes ce qui nécessite d'anonymiser ces informations avant de les partager. Plusieurs techniques d'anonymisation existent : la généralisation, la permutation, la perturbation et l'agrégation. Des modèles de protection de la vie privée sont validés via les techniques d'anonymisation : k-anonymat et l-diversité. Afin de garder la possibilité du profilage et maximiser les traitements possibles sur les données, la technique choisie dans ce projet est la généralisation.

mail	naissance	ville	heure	lang
jean@live.com	1981/03/08	Paris	2020/01/25 :15	FR
toto@gmail.com	1981/11/24	Levallois	2020/01/25 :13	EN
ali@hassan.sy	1981/08/08	Levallois	2020/01/25 :14	AR
jean@live.com	1981/03/08	Paris	2020/01/26 :09	FR
toto@gmail.com	1981/11/24	Levallois	2020/01/26 :10	EN
franck@live.fr	1981/04/30	Paris	2020/01/26 :11	FR
soso@gmail.com	1981/05/03	Grenoble	2020/02/29 :20	RU

TAB. 1 – *Extrait des données*

Ce papier est structuré de la façon suivante : La section 2 analyse l'état de l'art. La section 3 présente les définitions et les formalisations. Les sections 4 et 5 détaillent respectivement notre proposition de pseudonymisation et d'anonymisation. Nous concluons en section 6.

2 Positionnement et État de l'art

Dans la littérature, le chiffrement et le hachage sont utilisés afin de réaliser la pseudonymisation. Le chiffrement est généralement appliqué au niveau tuple. Des méthodes de chiffrement basées sur l'indexation stockent des index avec les tuples chiffrés (Mykletun et Tsudik, 2006). Ces méthodes permettent d'effectuer des agrégations et des requêtes sur les tuples chiffrés. Le chiffrement homomorphe est également utilisé dans ce contexte de protection de DCP. Ce chiffrement permet d'effectuer des opérations arithmétiques arbitraires sur des données chiffrées sans déchiffrement tout en offrant une sécurité sémantique (Gentry, 2009). Les auteurs de (Ribeiro et Nakamura, 2019) traitent des données personnelles critiques liées à l'obésité des enfants. Ils proposent une solution de pseudonymisation des identifiants des enfants et des parents en utilisant une fonction de hachage. Ce qui protège les DCP mais garde un lien entre les IDs des enfants et les IDs des parents.

Par contre, aucun de ces travaux ne prend en compte la différence entre les catégories des DCP (identifiant, quasi-identifiant et attribut sensible). Plus précisément, Ils ne proposent pas un traitement (chiffrement/hachage) spécifique pour chaque catégorie.

La généralisation dans le Big data est très bien traitée dans la littérature dans des nombreux travaux : avec Hadoop (Nandini Prasaad K.S. et Pratheek T.R., 2015) , avec Spark (Canbay et Sağiroğlu, 2017) et avec MapReduce (Zhang et al., 2014). Par contre, tous ces travaux dépendent de la technique utilisée. Autrement dit, ils ne sont pas réutilisables de manière générique avec d'autres technologies. En outre, ils nécessitent d'analyser les données à généraliser

plusieurs fois (plusieurs parcours) ce qui est compliqué et coûteux dans le Big data. Pour pallier cela, nous proposons une généralisation à une seule étape (un seul parcours des données). Nous proposons également une deuxième étape de validation qui filtre les données qui ne respectent pas les modèles de protection de la vie privée k-anonymat et l-diversité.

Nous avons choisi d'implémenter notre solution de pseudonymisation et d'anonymisation sous forme de plugins en Logstash. Ce choix est justifié par les raisons suivantes : (1) Logstash est un système ETL (Extract-transform-load) en temps réel indépendant de techniques de stockage de données (2) il est adapté au Big Data (3) il peut être exécuté en parallèle et il est évolutif (scalable), ce qui veut dire que l'on peut ajouter des nouveaux serveurs Logstash pour réaliser la même tâche en cas de volume de données à traiter accru par exemple.

3 Formalisations et définitions

Nous présentons ici les définitions permettant de formaliser une base de données.
Un **attribut (**att_i**)**, est une séquence nommée et typée des valeurs $\langle V_{att_i}, N_{att_i}, T_{att_i} \rangle$. Où : $V_{att_i} = (v_0, v_1, ..., v_j)$ est le vecteur des valeurs, N_{att_i}, T_{att_i} sont le nom et le type de l'attribut.

Une **collection de données (D)** est un ensemble d'attributs $\{att_1, att_2, \ldots, att_n\}$. Les attributs d'une collection de données D sont organisés dans les quatre groupes suivants :

- un **identifiant (ID)** est un ensemble d'attributs qui permettent d'identifier une personne directement. Par exemple, nom, mail, etc. ID $\subseteq$ D.
- un **quasi-identifiant (QID)** est un ensemble d'attributs qui permettent d'identifier une personne indirectement. Par exemple, la date de naissance, la ville natale. QID $\subseteq$ D.
- un **attribut sensible (AS)** est un ensemble d'attributs qui font apparaître, les origines raciales ou ethniques, les opinions politiques, philosophiques ou religieuses des personnes ou données relatives à la santé des personnes. AS $\subseteq$ D.
- **(NDCP)** est un ensemble d'attributs qui ne sont pas à caractère personnel. NDCP $\subseteq$ D.

On pose les données à caractère personnel **DCP** $= ID \cup AS \cup QID$. D=$DCP \cup NDCP$.

Un **Cryptosystème (C)** est un système qui permet la transformation d'un vecteur de valeurs $V_{att_i} = (v_0, v_1, ..., v_j)$ en un vecteur chiffré $Vc_{att_i} = (vc_0, vc_1, ..., vc_j)$. Il garantit une confidentialité optimale à l'utilisateur ç-à-d l'information dans la collection de données D ne peut pas être lue par des personnes non autorisées. Le chiffrement et le déchiffrement nécessitent une ou plusieurs clés secrètes connues uniquement par les personnes autorisées.

4 Pseudonymisation des DCP

La pseudonymisation est utile pour protéger les données stockées dans la mesure où il n'est pas toujours possible de rendre les données anonymes. Elle est également utile pour conserver les informations nécessaires à des fins de traitement : scientifiques, statistiques ou même historiques. Le chiffrement est couramment utilisé afin de réaliser ces objectifs. La pseudonymisation par chiffrement remplace les attributs identifiants (ID) et quasi-identifiants (QID) par d'autres chiffrés ce qui signifie que l'identité est cachée, mais en même temps la possibilité de la ré-identifier reste possible. Dans ce sens, nous implémentons pour notre projet un plugin qui permet de pseudonymiser les données via le chiffrement.

4.1 Chiffrement via un plugin Logstash

Notre plugin présenté dans cette section est basé sur un système de chiffrement symétrique (AES256). Pour adapter notre proposition au contexte RGPD, nous proposons un processus de chiffrement spécifique pour chaque catégorie de DCP (ID, QID et AS). Comme présente la Figure 1(a), ce plugin prend deux types d'entrées :

— la classification des catégories pour les DCP : ID, QID et AS, ainsi que les valeurs des données associées à chaque catégorie.
— les paramétrages de chiffrement (deux clés).

Nous supposons que les identifiants (ID) nécessitent plus de sécurité que les quasi-identifiants (QID) et que les attributs sensibles (AS). Donc, notre plugin effectue deux chiffrements différents (avec deux clés différentes) pour protéger les QID et les AS. Pour les QID : $C(QID) = C_{k1}(QID)$ et pour les AS : $C(AS) = C_{k2}(AS)$. Pour protéger les ID, notre plugin exécute les deux chiffrements en cascade (double chiffrement) : $C(ID) = C_{k2}(C_{k1}(ID))$. Ainsi, dans le chiffrement double nous utilisons deux clés secrètes.

Un changement régulier des clés est recommandé. Les données non personnelles NDCP ne sont pas chiffrées. Un exemple (extrait du fichier de configuration de Logstash) d'utilisation du plugin de chiffrement est présenté ci-dessous :

```
1: chiffrement {
2:    ID => ["mail"]
3:    QID => ["naissance", "ville", "heure"]
4:    AS => ["lang"]
5:    K1 => "Cle1"
6:    K2 => "Cle2"}
```

4.2 Déchiffrement via un plugin Logstash

Par ailleurs, le contexte de notre projet nécessite de récupérer les valeurs d'origine pour certains traitements. Pour cette raison, nous avons implémenté un plugin de déchiffrement. Ce plugin se base sur des signatures afin d'effectuer le traitement de déchiffrement. Nous avons intégré ces signatures dans le plugin de chiffrement présenté ci-dessus. Leur rôle est de distinguer le chiffrement utilisé parmi les trois mis en place (C_{K1}, C_{K2} et $C_{K2} \circ C_{K1}$). Le plugin de déchiffrement prend en entrée les vecteurs chiffrés (Vc_{att_i}) avec leurs types d'origine (T_{att_i}). Il retransforme les valeurs déchiffrées en leurs types d'origine. Un exemple d'utilisation du plugin de déchiffrement est ci-dessous :

```
1: dechiffrement {
2:    DCPC => [{"mail"=>"Texte"}, {"naissance""=>"Date"}, {"ville""=>"Texte"},
3:             {"heure""=>"Date"}, {"lang""=>"Texte"}]
4:    K1 => "Cle1"
5:    K2 => "Cle2"}
```

5 Anonymisation des DCP

L'anonymisation des DCP consiste à modifier les valeurs et/ou les attributs afin de rendre impossible la ré-identification des identités des personnes concernées. L'anonymisation est en particulier utilisée pour la diffusion et le partage de données d'intérêt public, comme les données ouvertes (Open data). Afin de mettre en oeuvre cette anonymisation, nos travaux se sont

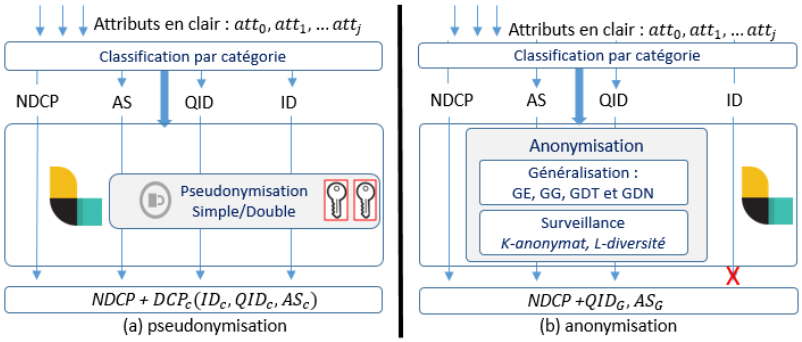

FIG. 1 – *Découverte des catégories et (a) pseudonymisation (b) anonymisation*

basés sur la technique de généralisation. Les modalités de généralisation ont été sélectionnées selon le type et les valeurs des données. Dans ce papier, nous proposons un plugin Logstash afin d'implémenter plusieurs formes de généralisation.

Afin de valider l'efficacité de la généralisation (niveau de protection et qualité de données), il est nécessaire de vérifier des paramètres des modèles de protection (k-anonymat, l-diversité). Un deuxième plugin de surveillance est mis en place pour effectuer cette vérification (cf. Figure 1(b)). Dans la suite de cette section, nous détaillons ces deux plugins d'anonymisation.

5.1 Plugin généralisation

Notre plugin consiste à supprimer les ID et appliquer la généralisation sur les QID et les AS (cf. Figure 1(b)). Les différentes modalités de généralisation sont modélisées par plusieurs types de règles. Elles sont détaillées dans la suite à l'aide de l'exemple suivant :

```
1: generalisation {
2:    regles => {
3:        "lang"  => ["RU","AR", "Autre"]
4:        "ville"  => ["Paris", "Levallois", "IDF"]
5:        "ville"  => ["Grenoble", "Lyon", "RA"]
6:        "naissance"  => "yyyy"
7:        "heure"  => [0, 6, 12, 18, 24]}}
```

5.1.1 Généralisation élémentaire (GE)

Elle est applicable sur plusieurs catégories de données. La règle qui définit la généralisation élémentaire est la suivante :

$$regle_x : N_{att_i} => [V_j \subseteq V_{att_i}, V_{G_i}] \quad (1)$$

Où V_j est le vecteur des valeurs à généraliser, V_{G_i} est la valeur de généralisation, $V_{G_i} \in H_i$ et H_i est une hiérarchie de généralisation de l'attribut att_i. Chaque valeur de V_j appartient à la même hiérarchie H_i à condition que le niveau de V_{G_i} soit supérieur au niveau des valeurs de V_j. Cette règle permet de remplacer les valeurs dans le vecteur V_j par la valeur V_{G_i}.

Ce type de généralisation permet de remplacer les valeurs rares d'un attribut par une seule valeur. Dans ce cas, une règle unique par attribut est suffisante. Une maîtrise des données

et une bonne connaissance des valeurs rares dans les données sont nécessaires afin de garantir une meilleure performance de ce plugin. Par exemple, la règle de génération de l'attribut "lang" (ligne 3) remplace les langues rares dans la Table 1 ("RU" et "AR") par la valeur ("Autre").

5.1.2 Généralisation Générique (GG)

C'est un cas général de GE (généralisation élémentaire). La même règle présentée dans Équation 1 définit la généralisation générique. Ce type de généralisation permet de remplacer toutes les valeurs d'un attribut par des valeurs plus génériques. Pour faire cela plusieurs règles par attribut sont nécessaires. Par exemple, les règles de généralisation de l'attribut "ville" (lignes 4 et 5) remplacent toutes les villes dans la Table 1 par l'abréviation de leur région. Une maîtrise des données est nécessaire et permet de tirer pleinement partie des possibilités de paramétrages qu'offre le fichier de configuration de ce plugin.

5.1.3 Généralisation datetime par niveau (GDN)

C'est une opération de généralisation spécifique pour les données de type date-temps. La règle qui définit la généralisation GDN est la suivante :

$$regle_x : N_{att_i} => N_{G_i} | N_{G_i} \in \{"yyyy", "mmm", "dd", "hh", "mm", "ss"\} \quad (2)$$

Via ce type de généralisation, nous offrons la possibilité de généraliser les données temporelles sur six niveaux : *"ss", "mm", "hh", "dd", "mmm" et "yyyy"* : seconde, minute, heure, jour, mois et année respectivement. Par exemple, la règle de génération de l'attribut "naissance" (ligne 6) remplace les dates de naissance de la Table 1 par les années de naissance.

5.1.4 Généralisation datetime par tranche d'heure (GDT)

C'est une deuxième façon de généraliser les données temporelles : le regroupement des données temporelles par tranches d'heures. La règle qui définit la généralisation GDT est :

$$regle_x : N_{att_i} => [00h, h_1, .., h_i, .., 24h] | h_j \in [1, 23] \wedge 00h < h_1 < h_i < 24h \quad (3)$$

Cette règle est paramétrée par une liste des heures ordonnée qui commence par l'heure 00h et se termine par l'heure 24h (afin de couvrir les 24h d'une journée). Cette liste doit contenir au moins 3 valeurs (2 intervalles de temps). Toutes les données temporelles qui appartiennent à un intervalle seront remplacées par la borne inférieure de l'intervalle. Par exemple toutes les valeurs dans l'intervalle $[00h, 1h]$ seront remplacées par $00h$. Dans notre exemple, la règle de la généralisation de l'attribut "heure" (ligne 7) remplace les heures des données temporelles par les heures présentées dans cette liste $[0, 6, 12, 18, 24]$.

La Table 2 présente le résultat d'anonymisation effectuée via notre plugin de généralisation selon la configuration présentée dans l'exemple précédent sur les données de la Table 1.

5.2 Plugin de surveillance

Le plugin de surveillance valide deux modèles de protection de la vie privée : k-anonymat Samarati (2001), l-diversité Machanavajjhala et al. (2006). Le modèle k-anonymat protège les

classe	naissance	ville	heure	lang
1	1981	IDF	2020/01/25 :12	FR
1	1981	IDF	2020/01/25 :12	EN
1	1981	IDF	2020/01/25 :12	Autre
2	1981	IDF	2020/01/26 :06	FR
2	1981	IDF	2020/01/26 :06	EN
2	1981	IDF	2020/01/26 :06	FR
3	1981	RA	2020/02/29 :18	Autre

TAB. 2 – *Données anonymisées (avant validation par le plugin de surveillance)*

identités en validant qu'il y a au moins (k) tuples (individus) qui ont les mêmes valeurs de QID généralisées. Ces tuples forment une classe d'équivalence. Le modèle l-diversité renforce la protection de k-anonymat en validant la diversité des valeurs des attributs sensibles dans chaque classe d'équivalence. Autrement dit, le modèle l-diversité nécessite d'avoir au moins (l) valeurs différentes pour chaque attribut sensible dans chaque classe d'équivalence.

Dans ce plugin nous effectuons une analyse sur les données anonymisées via le plugin de généralisation. Ce plugin prend en entrées les paramètres des modèles k-anonymat et l-diversité. Il ne retourne que l'ensemble des tuples qui respectent ces modèles. Les paramètres du modèle k-anonymat sont la valeur de *(k)* et l'ensemble des attributs qui déterminent la classe d'équivalence. De la même manière, les paramètres du modèle l-diversité sont la valeur de *(l)* et l'ensemble des attributs sensibles dont le plugin doit vérifier la diversité de leurs valeurs.

Une mémoire commune entre les différents threads de Logstash est utilisée, afin de pouvoir exécuter le plugin de surveillance en parallèle. Dans cette mémoire, nous stockons les tuples des classes en cours de validation. Une fois qu'une classe est validée, ses tuples sont envoyés dans la sortie de ce plugin et l'espace mémoire utilisé est libéré.

Exemple : Pour valider le résultat de la généralisation présentée dans la Table 2, nous avons sélectionné les attributs ("naissance", "ville", "heure") pour définir la classe d'équivalence pour notre plugin de surveillance. Le modèle l-diversité est appliqué sur un seul attribut sensible ("lang"). Pour cet exemple, nous avons choisi 3-anonymat et 3-diversité. La configuration du plugin de surveillance correspondant à cet exemple est présenté ci-dessous :

```
1: surveillance {
2:    K => 3
3:    K_classe => ["naissance", "ville", "heure"]
4:    L => 3
5:    L_attributs => ["lang"]}
```

En considérant cette configuration, la Table 2 contient trois classes d'équivalence. La première classe respecte le 3-anonymat parce qu'il y a trois tuples qui ont les mêmes valeurs des QID. Elle respecte également le 3-diversité parce qu'elle contient trois valeurs différentes de l'attribut sensible "lang". La deuxième classe respecte le 3-anonymat, par contre il n'y a que deux valeurs différentes de "lang" donc elle ne respecte pas le 3-diversité. La troisième classe ne contient qu'un seule tuple donc elle ne respecte pas le 3-anonymat. Pour conclure, le résultat final de l'anonymisation comprend seulement la classe 1 de la Table 2.

6 Conclusion

Nous avons présenté dans ce papier des plugins Logstash afin de renforcer la sécurité sur les données dont l'objectif est de répondre aux besoins de conformité RGPD. Notre solution

de pseudonymisation est basée sur un système de chiffrement symétrique. Nous appliquons un chiffrement simple sur les attributs de type QID et AS avec deux clés différentes. Un chiffrement double en cascade est appliqué sur les attributs de type ID. Nous avons proposé un plugin de déchiffrement afin de retrouver les valeurs d'origine des attributs.

Notre solution d'anonymisation met en oeuvre plusieurs techniques de généralisation : (1) **une généralisation élémentaire** pour traiter les valeurs rares d'un attribut, (2) **une généralisation générique** afin d'anonymiser toutes les valeurs des attributs et deux techniques de généralisation spécifiques pour les données temporelles : (3) **une généralisation par niveau temporel** et (4) **une généralisation par tranche d'heure**. En outre, nous proposons un plugin de validation de la qualité de protection en respectant les modèles *k-anonymat* et *l-diversité*.

Références

Canbay, Y. et S. Sağiroğlu (2017). Big data anonymization with spark. *2nd International Conference on Computer Science and Engineering, UBMK 2017*, 833–838.

Gentry, C. (2009). *A Fully Homomorphic Encryption Scheme*. Ph. D. thesis, Stanford, USA.

Machanavajjhala, A., J. Gehrke, D. Kifer, et M. Venkitasubramaniam (2006). L-diversity : privacy beyond k-anonymity. In *22nd International Conference on Data Engineering*.

Mrabet, A., A. Hassan, et P. Darmon (2019). Détection des données à caractère personnel dans les bases multidimensionnelles. In *EDA*, Volume B-15 of *RNTI*, pp. 31–44.

Mykletun, E. et G. Tsudik (2006). Aggregation queries in the database-as-a-service model. In *Data and Applications Security XX, Proceedings*, Volume 4127 of *LNCS*, pp. 89–103.

Nandini Prasaad K.S. et Pratheek T.R. (2015). Providing anonymity using top down specialization on big data using hadoop framework. In *IEEE India Conference*, pp. 1–6.

Ribeiro, S. L. et E. T. Nakamura (2019). Privacy protection with pseudonymization and anonymization in a health iot system : Results from ocariot. In *IEEE 19th International Conference on Bioinformatics and Bioengineering*, pp. 904–908.

Samarati, P. (2001). Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering 13*(6), 1010–1027.

Zhang, X., C. Liu, S. Nepal, C. Yang, W. Dou, et J. Chen (2014). A hybrid approach for scalable sub-tree anonymization over big data using MapReduce on cloud. *Journal of Computer and System Sciences 80*(5), 1008–1020.

Summary

This paper presents a new implementation of personal data protection methods based on specific encryption and generalization algorithms implemented in Logstash plugins. Our encryption algorithm is adapted to personal data by considering the different categories: identifiers, quasi-identifiers, and sensitive attributes. In addition, our anonymization solution offers several generalization methods, which can be configured according to the data type. To validate the results of these methods, we also propose a step of verifying privacy protection models such as k-anonymity and l-diversity.